

## [0149] CLAIMS

What is claimed is:

1. A method comprising:

receiving an instruction to open an eXtensible Markup Language (XML) document;

searching the XML document to locate a processing instruction (PI) containing a href attribute that points to a URL;

discovering a solution using the URL in the PI;

opening the XML document with the solution, wherein:

the solution includes an extensible stylesheet language (XSLT) presentation application and a XML schema;

the XML document can be inferred from the XML schema; and

portions of the XML document are logically coupled with fragments of the XML schema;

executing the XSLT presentation application to render a Hypertext Markup Language (HTML) electronic form containing data-entry fields associated with the coupled portions.

2. The method as defined in Claim 1, wherein one or more of the receiving, the searching, the examining, the discovering, the opening, and the executing of the XSLT presentation application are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

3. The method as defined in Claim 1, wherein:  
the executing the XSLT presentation application comprises applying an XSLT stylesheet to the XML document to create the HTML electronic form; and  
the HTML electronic form includes a plurality of HTML elements corresponding to the data-entry fields.

4. The method as defined in Claim 1, wherein:  
the searching the XML document further comprises finding the first PI in the XML document; and  
the first PI in the XML document includes the URL.

5. The method as defined in Claim 1, wherein the solution further comprises a manifest of all files that can be used for:  
representing the XML document in the HTML electronic form;  
allowing a user in input data into the data-entry fields; and  
validating the data that the user inputs into the data-entry fields.

6. The method as defined in Claim 1, wherein the coupled portions contain information setting forth all possible XML documents for the coupled portions.

7. The method as defined in 1, wherein:  
the data-entry fields of the HTML electronic form map to a corresponding plurality of nodes of the XML document; and  
the method further comprises:  
receiving, through the data-entry fields, data input by a user for storage in a corresponding said node in the XML document; and  
outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.

8. The method as defined in Claim 7, wherein:  
the XML schema includes a logic application; and  
the method further comprises:  
executing the logic application to perform a validation to determine if the data received by input from the user is valid or invalid; and  
when the validation determines that the data received by input from the user is invalid, outputting a dialog box bearing indicia informing the user that the data input is invalid.

9. The method as defined in Claim 8, wherein:

the validation is performed on the data received by input from the user into each said data-entry field with a validation rule;

the logic application comprises a plurality of the validation rules for:

a corresponding plurality of the nodes in the XML document; and

a corresponding plurality of the data-entry fields;

the validation uses each said validation rule to:

determine if the data received by input from the user into a corresponding said data-entry field is valid or invalid; and

require the user to correct any data input into the corresponding said data-entry field that the validation determines to be invalid.

10. The method as defined in Claim 9, wherein each said validation rule has an identity that is selected from the group consisting of:

the identity is based on a part of a schema governing a corresponding said node;

the identity is written in script and associated with a corresponding said node; and

the identity is written in a declarative syntax and associated with a corresponding said node.

11. The method as defined in Claim 9, wherein:

each said validation rule includes an alert area display; and

the validation further comprises:

using one said validation rule to determine that the data received by input from the user into a corresponding said data-entry field is invalid;  
and

outputting the corresponding alert area display so as to be associated with the corresponding said data-entry field.

**12.** The method as defined in Claim 11, wherein when the alert area display is output, the output includes one or more characteristics selected from the group consisting of:

- graphics surrounding the corresponding said data-entry field;
- the alert area display surrounds the corresponding said data-entry field;
- the alert area display includes graphics containing a red, dashed-lined box;
- the alert area display includes graphics highlighting the data in the corresponding said data-entry field;
- the alert area display surrounds the corresponding said data-entry field and includes the graphics containing a squiggly line beneath the data in the corresponding said data-entry field;
- the alert area display includes text containing information about the invalid data in the corresponding said data-entry field;
- the alert area display includes text containing information about the corresponding said data-entry field; and
- the alert area display includes a pop-up window.

**13.** The method as defined in Claim 9, wherein each said node has one or more of the validation rules associated therewith.

**14.** The method as defined in Claim 9, wherein one said validation rule includes a requirement that is selected from the group consisting of:

the data received by input from the user into a corresponding said data-entry field is to be within a certain range;

the data received by input from the user into a corresponding said data-entry field is to be within a certain range of text or numerals for a setting of one or more bounds of the certain range;

the data received by input from the user into a corresponding said data-entry field is to be numerical;

the data received by input from the user into a corresponding said data-entry field is to be textual; and

the data received by input from the user into a corresponding said data-entry field that references another said node in the data file.

**15.** The method as defined in Claim 9, wherein the plurality of the validation rules are associated by mapping to the corresponding plurality of the nodes in the data file.

**16.** The method as defined in Claim 9, wherein each said validation rule is associated by mapping to a corresponding said data-entry field by use an entity selected from the group consisting of an XPath expression, a declarative syntax, and an entity that is script-based.

**17.** The method as defined in Claim 16, wherein the script-based validation rule maps to a corresponding said node with an entity selected from the group consisting of:

- an XPath expression;

- an event handler;

- an event handler that determines when a real-time validation tool uses the script-based validation rule;

- an event handler that determines when a real-time validation tool uses the script-based validation rule before data received for the node is held by the data file; and

- an event handler that determines when a real-time validation tool uses the script-based validation rule after data received for the node is held by the data file.

**18.** The method as defined in Claim 9, wherein each said validation rule includes:

- an alert area display; and

- how the alert area display is to appear when output.

**19.** The method as defined in Claim 1, wherein the PI includes a character string of “mso-InfoPathSolution”.

**20.** A computer-readable medium comprising instruction that, when executed by a computer, performs the method of Claim 1.

**21.** A method comprising:

- receiving an instruction to open a XML document;
- searching the XML document to locate a processing instruction (PI) having a name;
- examining the name of the PI to assess the likelihood that the PI includes a solution identifier for the solution; and
- when the likelihood exceeds a threshold, discovering a solution using the name in the PI;
- opening the XML document with the solution, wherein:
  - the solution includes a XSLT presentation application and a XML schema;
  - the XML document can be inferred from the XML schema; and
  - portions of the XML document are logically coupled with fragments of the XML schema;
- executing the XSLT presentation application to render an HTML electronic form containing data-entry fields associated with the coupled portions.



**22.** The method as defined in Claim 21, wherein one or more of the receiving, the searching, the examining, the opening, and the executing of the XSLT presentation application are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

**23.** The method as defined in 21, wherein:

the data-entry fields of the HTML electronic form map to a corresponding plurality of nodes of the XML document; and

the method further comprises:

receiving, through the data-entry fields, data input by a user for storage in a corresponding said node in the XML document; and

outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.

**24.** The method as defined in Claim 21, wherein the PI includes a character string of “mso-InfoPathSolution”.

**25.** A computer-readable medium comprising instruction that, when executed by a computer, performs the method of Claim 21.

**26.** A method comprising:

receiving an instruction to open a XML document;

searching the XML document to locate a processing instruction (PI) having a target that includes the character string that identifies an application used to create an HTML electronic form associated with the XML document;

examining one of a URL or an URN in the PI to assess the likelihood that the PI includes a solution identifier for the solution; and

when the likelihood exceeds a threshold, discovering a solution using the one of a URL or an URN;

opening the XML document with the solution, wherein:

the solution includes a XSLT presentation application and a XML schema;

the XML document can be inferred from the XML schema; and

portions of the XML document are logically coupled with fragments of the XML schema;

executing the XSLT presentation application to render the HTML electronic form containing data-entry fields associated with the coupled portions.

**27.** The method as defined in 26, wherein:

the data-entry fields of the HTML electronic form map to a corresponding plurality of nodes of the XML document; and

the method further comprises:

receiving, through the data-entry fields, data input by a user for storage in a corresponding said node in the XML document; and

outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.

**28.** The method as defined in Claim 26, wherein the character string is “mso-InfoPathSolution”.

**29.** The method as defined in Claim 26, wherein one or more of the receiving, the searching, the examining, the discovering, the opening, and the executing of the XSLT presentation application are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

**30.** The method as defined in Claim 26, wherein:  
the assessment of the likelihood exceeds the threshold when the PI is the first PI in the XML document that contains a URL is the first URL; and  
the discovering a solution comprises using the first URL to discover the solution.

**31.** The method as defined in Claim 26, wherein:  
the assessment of the likelihood exceeds the threshold when the one of a URL or an URN is a URL having a path with a suffix that is selected from the group consisting of “.xsf” or “.xsn”.

**32.** A computer-readable medium comprising instruction that, when executed by a computer, performs the method of Claim 26.

**33.** A method comprising:

- receiving an instruction to open a XML document;
- searching the XML document to locate a processing instruction (PI) having a href attribute and at least one of a PI version and a product version;
- discovering a solution using a name associated with the href attribute;
- opening the XML document with the solution, wherein:
  - the solution includes a XSLT presentation application and a XML schema;
  - the XML document can be inferred from the XML schema; and
  - portions of the XML document are logically coupled with fragments of the XML schema;
- executing the XSLT presentation application to render an HTML electronic form containing data-entry fields associated with the coupled portions.

**34.** The method as defined in Claim 33, wherein one or more of the receiving, the searching, the examining, the discovering, the opening, and the executing of the XSLT presentation application are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

**35.** The method as defined in Claim 33, wherein:

- the data-entry fields of the HTML electronic form map to a corresponding plurality of nodes of the XML document; and
- the method further comprises:

receiving, through the data-entry fields, data input by a user for storage in a corresponding said node in the XML document; and

outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.

**36.** A computer-readable medium comprising instruction that, when executed by a computer, performs the method of Claim 33.

**37.** A method comprising:

receiving an instruction to open a XML document;

searching the XML document to locate a processing instruction (PI) having a href attribute and at least one of a PI version and a product version;

discovering a solution using a name in the PI that is associated with the href attribute;

opening the XML document with the solution, wherein:

the solution includes a XSLT presentation application and a XML schema;

the XML document can be inferred from the XML schema; and

portions of the XML document are logically coupled with fragments of the XML schema;

executing the XSLT presentation application to render an HTML electronic form containing data-entry fields associated with the coupled portions.

**38.** The method as defined in Claim 37, wherein one or more of the receiving, the searching, the discovering, the opening, and the executing of the XSLT presentation application are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

**39.** The method as defined in 37, wherein:

the data-entry fields of the HTML electronic form map to a corresponding plurality of nodes of the XML document; and

the method further comprises:

receiving, through the data-entry fields, data input by a user for storage in a corresponding said node in the XML document; and

outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.

**40.** A computer-readable medium comprising instruction that, when executed by a computer, performs the method of Claim 37 .

**41.** A computer-readable medium including instructions that, when executed by a computer, perform acts comprising:

receiving an instruction to open an XML document;

searching the XML document to locate a processing instruction (PI) that contains an entity selected from the group consisting of:

a href attribute that points to a URL;

a name;

a target that includes the character string that identifies an application used to create an HTML electronic form associated with the XML document; and

a href attribute and at least one of a PI version and a product version;

discovering a solution using the entity in the PI;

opening the XML document with the solution, wherein:

the solution includes an XSLT presentation application and an XML schema;

the XML document can be inferred from the XML schema; and

portions of the XML document are logically coupled with fragments of the XML schema;

executing the XSLT presentation application to transform the coupled portions of the XML document into an HTML electronic form containing data-entry fields associated with the coupled portions.

**42.** The computer-readable medium as defined in Claim 41, wherein one or more of the acts are performed by the execution of an HTML electronic forms application that is different from the application used to create the HTML electronic form.

**43.** The computer-readable medium as defined in Claim 41, wherein the discovering a solution using the entity in the PI comprises an act selected from the group consisting of:

(i) discovering the solution using the URL in the PI;

(ii) examining the name of the PI to assess the likelihood that the PI includes solution identifier for the solution; and

when the likelihood exceeds a threshold, discovering the solution using the name in the PI;

(iii) examining one of a URL or an URN in the PI to assess the likelihood that the PI includes a solution identifier for the solution; and

when the likelihood exceeds a threshold, discovering the solution using the one of a URL or an URN;

(iv) discovering the solution using a name associated with the href attribute;

(v) discovering the solution using a name in the PI that is associated with the href attribute; and

(vi) a combination of the foregoing.

**44.** The computer-readable medium as defined in Claim 41, wherein:  
the data-entry fields of the HTML electronic form map to a corresponding plurality of nodes of the XML document; and

the acts further comprise:

receiving, through the data-entry fields, data input by a user for storage in a corresponding said node in the XML document; and



outputting data in XML for viewing by the user in the HTML electronic form through the data-entry fields via the mapping of the data-entry fields from corresponding said nodes of the XML document.

**45.** The computer-readable medium as defined in Claim 41, wherein the character string is “mso-InfoPathSolution”.